

WEST

Generate Collection

Print

L7: Entry 9 of 19

File: PGPB

Sep 19, 2002

DOCUMENT-IDENTIFIER: US 20020131565 A1

TITLE: Calendaring systems and methods

Abstract Paragraph (1):

A calendaring system communicatively coupled to a network comprises: a calendar engine capable to store and display event data from a calendar database; a portrait database capable to store portraits of users; the portraits including relationship settings for users; and an event engine, communicatively coupled to the calendar engine and portrait database, capable of scheduling events (including implicit events), sending events invitations and responding to an event invitation received, and offering appropriate services related to those events, via the network, as a function of time availability as indicated in the calendar database, relationship settings, and lifestyle wishes, monitors and gauges of the participants in the event as indicated in the portrait database.

Summary of Invention Paragraph (4):

[0003] Conventional electronic calendars enable users to schedule events, such as meetings, telephone conferences, birthday parties, etc. In addition, some conventional electronic calendars enable a user to invite other users, to send invitations to events and receive responses such as "accept", "decline", and "reschedule". The only kind of interaction the calendars provide is to give you notification of event proximity. The calendars do not provide intelligent decision-making, i.e. they will let you schedule back-to-back meetings in two different countries. Everything that the calendar stores is explicitly provided by somebody and they do not take into account implicit events related to explicit (e.g. travel time) and implicit states of mind (e.g. stress level . . . the amount of time needed to prepare for an event). A new method is needed to truly add a level of convenience to people's busy scheduling needs.

Summary of Invention Paragraph (23):

[0021] As an example, several somewhat stressful events scheduled back-to-back without relief will create a sum total of higher stress into a user's "stress-o-meter." Also, one intensely romantic evening might suffice to meet a user's desired setting on the "love-o-meter." The portrait gallery engine maintains contact information and/or profiles for other users. A user may enter data about users into a portrait gallery database. Alternatively, or in addition, the engine may import contacts from Outlook and/or vCards from email or other data acquisition techniques. A user can define contacts by including such information as type of living organism (e.g., adult; dependent adult; child; dependent child; baby; animal/pet; etc.), the user's emotional relationship to the contact, and traditional relationship (romantic, friend; business colleague; etc.). In addition, the portrait gallery engine enables the user to form groups of contacts and define information about the groups similarly to defining information about individual contacts.

Summary of Invention Paragraph (27):

[0025] The call scheduler engine, also a Triggered Service, works similarly to the voicemail event engine in that it enables a user to schedule a telephone conference with multiple invitees and reschedule if an invitee isn't available to make the conference. The rescheduling can be based on criteria such as a user's preference and upon invitees' availability according to their calendars. The call scheduler also allows attendees to attach files pertaining to the event (e.g. voice-mail, e-mail, and documents) to the attendees' calendars.

Summary of Invention Paragraph (29):

[0027] A second method comprises: scheduling an event or telephone conference; sending invitations to invitees; receiving responses from invitees; notifying the moderator (inviter) of the responses; receiving the inviter's determination regarding scheduling

of meeting based on the responses; notifying the invitees of cancellation if the inviter so determines; proposed rescheduling of the event by repeating the above steps; or dropping the invitee and continuing with scheduling method. If the invitee is dropped, the method further comprises notifying the invitee of being dropped. Whether or not the invitee is dropped, the method further comprises sending reminders to the remaining invitees; notifying the moderator to start the call; providing schedule status update to the invitees; notifying invitees to start the call at the scheduled time; receiving invitee availability; notifying the moderator about real time invitee availability; enabling the moderator to determine whether to cancel the call due to unavailability of some invitees, bump the call, or continue the call; starting the call; and sending a list of participants to all invitees or only call participants.

Brief Description of Drawings Paragraph (13):

[0052] FIG. 11 is a diagram illustrating an example GUI 1100 illustrating impact of delaying a scheduled telephone conference based on invitees' availability; and

Detail Description Paragraph (10):

[0062] Preferences 127 are set by a user and if set, enable life manager 125 to automatically schedule events in response to invitations from other users. Life manager 125, when receiving an invitation, first determines if it is allowed to accept or reject the invitation based on preferences 127. If the preferences 127 is set, then the life manager 125 determines the user's availability by examining the calendar database 117 for free time and also examining the user's portrait database 132 to examine the inviter's portrait, which will be discussed further below. For example, if a user receives an invitation for an event scheduled next Tuesday at 1 PM, the life manager 125 first determines if it is authorized to accept or decline the invitation by looking up preferences 127. If the life manager 125 is not authorized, the life manager 125 queries the user whether he or she wishes to accept. If the life manager 125 is authorized, then the life manager 125 first looks up the user's portrait for the inviter in database 132 to determine the nature of the relationship (e.g., friendly or not). If the portrait indicates that the relationship is acceptable, the life manager 125 then examines the user's schedule in calendar database 117 to determine if the user is available at the time of the event (e.g., next Tuesday at 1 PM). If the user is available, then the life manager 125 may accept the invitation, notify the inviter of the acceptance, and add the event to the user's calendar by updating database 117 to reflect the newly scheduled event.

Detail Description Paragraph (13):

[0065] The portrait gallery engine 130 maintains contact information for other users. A user may enter data about users into a portrait gallery database 132. Alternatively, or in addition, the engine 130 may import contacts from Outlook and/or vCards from email into database 132. A user can define contacts by including such information as type of living organism (e.g., adult; dependent adult; child; dependent child; baby; animal/pet; etc.), individual relationship (romantic, friend; business colleague; "time hog"; neutral, etc.), emotional relationships (e.g. "Which part of the restraining order don't you understand?!", "The Love of My Life, I Always Have Time for You," etc.), and time shown to the contact when the contact wants to schedule an event with the user. Defining contacts will be discussed in further detail in conjunction with FIG. 10. Alternatively, the individual relationship may be specified numerically with a high number representing an important person (e.g., boss, close friend, significant other), a low number representing an enemy (e.g., harasser), or a middle number (e.g., colleague). In addition, the portrait gallery engine 130 enables the user to form groups of contacts and define information about the groups similarly to defining information about individual contacts. Other engines, as discussed above and further below, use portraits in portrait gallery database 132.

Detail Description Paragraph (16):

[0068] The call scheduler engine 140 works similarly to the event engine 120 in that it enables a user to schedule a telephone conference with multiple invitees and reschedule if an invitee isn't available to make the conference. The rescheduling can be based on a user's preference or based on invitees' availability according to their calendars. The call scheduler engine 140 can also examine other users' calendars to determine availability for rescheduling. The call scheduler engine 140 may also limit rescheduling to the time shown preference set for the other users' portraits.

Detail Description Paragraph (18):

[0070] In an embodiment of the invention, consumer device 110 may also include a persistent java bar engine (not shown) that provides a floating Java application that allows enable calendar users quick access to important status information (such as the

number of received voicemails, emails and faxes. Short text messages such as headline news and sports and personal reminders and notes may also appear in the bar in a "news ticker" style. When invitations to join an event or telephone conference, the persistent java bar engine may display the invitation or an appropriate button to take the user to an RSVP page.

Detail Description Paragraph (25):

[0077] If the call does go to voicemail, then the appropriate answering machine message to play is determined (460). The answering machine messages are stored in answering machine messages 137 and can be correlated for specific callers and/or relationship types. For example, a user may have an answering machine message for a significant other and a more serious answering message for a colleague. In addition, a user may have an answering message that issues a warning to a caller having a negative relationship. The answering machine message may also enable the caller to schedule a callback based on the user's availability. The determined answering machine message is then played (470).

Detail Description Paragraph (26):

[0078] If it is determined (480) to give the caller an option to schedule a callback based on time shown for the caller in the user's portrait database 132 and the user's availability based on scheduled events in the user's calendar, scheduling information is received (490) from the caller via touchtone input or voice recognition and then the calendar database 117 is updated (495). In addition, the caller may also leave a voicemail. If the caller is not given the option to schedule a callback, the caller may leave a voicemail, which is recorded (485) and stored in stored voicemails 139. The method 400 then ends.

Detail Description Paragraph (28):

[0080] Scheduling data is then sent (532) to the invitees. At the appropriate time, the moderator (inviter) is notified (535) to start the call and a schedule status update is provided (537) that shows the impact of delaying the call (an example GUI 1100 showing the impact of delaying a call is shown in FIG. 11). For example, if an invitee is fully booked for the rest of the day, it will be hard to delay the start of the call. The invitees are then notified (540) to start the call. Invitee availability is then received (545) and the moderator is notified (547) of real time invitee availability. Based on this notification, the moderator may cancel (547) the call, at which point the invitees are notified (550) of such and the method 500 ends. Alternatively, the moderator may decide to move (552) the call, at which point the method 500 returns to sending (507). If the moderator decides not to cancel (547) and not to move (552), then the call is started (555) and a list of participants is sent (557) to all the participants and each participant now has access to associated files (559). The method 500 then ends.

Detail Description Paragraph (35):

[0087] FIG. 11 is a diagram illustrating an example GUI 1100 illustrating impact of delaying a scheduled telephone conference based on invitees' availability. The invitees' availability is shown in table 1110. The user can then determine to delay the call by pressing a delay button 1120, cancel the call by pressing a cancel button 1130, or start the call by pressing a start call button 1140. Further, if the user decides to delay the telephone conference, the user can specify the amount of the delay by setting 1125a and 1125b.

CLAIMS:

15. The method of claim 12, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

16. The method of claim 13, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

17. The method of claim 14, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

18. The method of claim 15, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call.

19. The method of claim 16, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call

20. The method of claim 17, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call.

26. The method of claim 24, wherein the determining includes accessing invitees' calendar databases to determine the invitees' availabilities.

27. The method of claim 24, wherein the determining includes accessing invitees' calendar databases, life style wishes, monitors and gauges, and relationship settings to determine the invitees' preferred time availabilities.

30. A system communicatively coupled to a network, comprising: a calendar engine capable to store and display event data from a calendar database; a portrait database capable to store portraits of users; the portraits including relationship settings for users; and an event engine, communicatively coupled to the calendar engine and portrait database, capable to respond to an event invitation received, via the network, from an inviter as a function of time availability as indicated in the calendar database and relationship setting of the invitee as indicated in the portrait database.

37. The system of claim 36, wherein the conference scheduler engine is capable to determine a time and date for a conference by determining availability of invitees by examining their respective calendar databases.

39. The system of claim 38, wherein the conference scheduler engine is further capable to display a schedule status update showing the impact of delaying a scheduled conference, wherein the update includes schedules of invitees as indicated in their respective calendar databases.

58. The computer-readable medium of claim 55, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

59. The computer-readable medium of claim 56, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

60. The computer-readable medium of claim 57, further comprising sending schedule availability information to the caller if the call is not enabled to ring through, the schedule availability information based on the caller relationship setting and a user's availability as indicated in a calendar database.

61. The computer-readable medium of claim 58, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call.

62. The computer-readable medium of claim 59, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call

63. The computer-readable medium of claim 60, further comprising: receiving a response to the sent schedule availability information, the response including a date and time for a telephone call; updating the calendar database to include the date and time for the telephone call.

WEST☐ **Generate Collection** **Print**

L7: Entry 10 of 19

File: PGPB

Jul 25, 2002

DOCUMENT-IDENTIFIER: US 20020099777 A1

TITLE: Integrating collaborative messaging into an electronic mail program

Summary of Invention Paragraph (6):

[0004] One possible solution to this problem is to store electronic mail content for a collaborative discussion on a centralized storage server (e.g., a web server or SQL server) and simply pass around a pointer to that storage in the electronic mail message. When a participant replies to the electronic mail the storage is updated and notification may be sent to the participants, which in turn can examine the latest status of the collaborative discussion by de-referencing the pointer to the storage sent around in the notification (or the original electronic mail). This can considerably simplify the ability to track discussions and facilitate collaborative decision making while at the same time avoiding electronic mail clutter.

Detail Description Paragraph (21):

[0045] Email and collaboration client 144 optionally includes multiple components: a calendar component 146, an email component 148, a contact manager component 150, and a task manager component 152. Calendar component 146 manages an electronic calendar for a user of client device 140. Information from calendar component 146 is made accessible to other components of client program 144, allowing those components to display indications to the user as to whether certain times are available in the user's schedule. For example, if the user receives an email message requesting a one-hour meeting at 3 pm on April 1, then email component 148 can communicate with calendar component 146 to determine whether the user has a conflicting appointment or meeting on April 1 and display such a conflict (or absence of a conflict) to the user along with the email message.

Detail Description Paragraph (23):

[0047] Contact manager component 150 manages one or more contact lists for the user of client device 140. The user can maintain various information for individuals or organizations, such as names, telephone numbers, mailing addresses, email addresses, etc., all of which is managed by component 150. The user can further create multiple different contact lists or address books to ease in organizing his or her contacts. Information from contact manager component 150 is also made accessible to other components of client program 144, allowing those components to take advantage of information already stored by component 150. By way of example, when a new collaborative email message is being created by a user, email component 148 may access contact manager component 150 to determine the email address or other personal information that corresponds to a particular recipient.

Detail Description Paragraph (41):

[0065] Email component 148 of FIG. 3 may also make a wide variety of additional options available to the author when creating a collaborative email message. The additional options may be selected by the author in any of a variety of manners, such as from a pull-down menu on menu bar 172, an icon on a toolbar 174, etc. Many of these options for collaborative email messages are the same as made available for conventional email messages. One such additional option is to give the author a "special" status compared to the recipients of the collaborative email message. This special status allows additional actions to be taken by the author that cannot be taken by the recipients, such as to require that the collaborative email message be encrypted or to prevent others (e.g., recipients) from modifying the collaborative email message (except, possibly, by answering questions in the collaborative email message or adding comments/feedback to the collaborative email message), the ability to "recall" the email message, etc. Any such actions taken by the author are identified in the collaborative email message and may be enforced by the email servers or the clients.

Detail Description Paragraph (44):

[0068] Additionally, email component 148 may present recipient (and possibly author) availability information to the user during creation of a new or response collaborative email message. This availability information may be displayed to the user in a variety of different manners, such as by displaying the users' addresses in fields 176 and/or 178 in different colors and underlines (e.g., green and solid for available, red and broken for unavailable). This availability information can refer to current availability (e.g., as the message is being created or responded to), or alternatively later availability (e.g., for a proposed meeting time).

Detail Description Paragraph (45):

[0069] The availability information can be obtained by email component 148 in a variety of different manners. Some information may be set manually by the user (e.g., "Out to Lunch", "Busy", etc.), while other information may be obtained via an intelligent sensing process. For example, to determine the availability of a particular recipient, email component 148 can communicate with its associated email server (or alternatively an instant messaging server), which in turn communicates with the server (email or instant messaging) associated with the recipient. Information regarding the recipient is obtained from the server associated with the recipient, such as the recipient's calendar to determine whether the recipient is currently in a meeting, or the recipient's current status online. If the recipient is currently in a meeting, then he or she is identified as being unavailable; otherwise, he or she is identified as being available. By way of another example, the recipient's telephone may be in communication with his or her client device, which in turn is in communication with his or her associated email server (or instant messaging server). Thus, the server is able to identify when the recipient is using the phone (e.g., receiver off the hook, phone line in use, etc.), so that he or she can be identified as unavailable if on the phone and otherwise available (or potentially available). This availability information can be used in a variety of different manners, such as allowing initiation of an instant messaging session that can subsequently be made part of a collaborative email message, informing the author of a collaborative email message who is available at the time the collaborative email message is created, etc.

Detail Description Paragraph (127):

[0149] Determining a good time to notify a particular user can be accomplished by considering one or more factors. Examples of such factors include the following. One factor is the importance of the topic to the user--this can be determined by the user previously identifying the importance, based on a category of the collaborative email message and other messages of the same category (whether collaborative or not) and how quickly the user reads those other messages. Another factor is who is the author of or the recipient that responded to the collaborative email message. Collaborative email messages (either new or replies) from certain individuals (e.g., the user's supervisor or other boss) may be viewed as more important to the user. Another factor is what the user is working on at the time client 140 becomes aware of the new/modified collaborative email message. If the user is working on a response to the same collaborative email message, or on another message or task with the same category, then the new/modified collaborative email message would be viewed as being more important to the user than collaborative email messages having other categories. Another factor is how busy the user currently is. For example, if the user is entering a lot of data (e.g., by typing, voice input, etc.) or having multimedia content rendered, he or she may be viewed as being busy with other tasks and thus notification of the new/modified collaborative email message would be less important. Another factor is a user-selectable notification status. For example, the user may have turned their notification status to "Busy", "Do not Disturb", "Out to Lunch", or some other setting that allows the user to be viewed as being busy with other matters and not to be disturbed by the notification of the new/modified collaborative email message.

CLAIMS:

16. A method as recited in claim 1, further comprising: comparing a time identified in the collaborative electronic mail message with an electronic calendar corresponding to the user; determining whether a conflict exists between the time identified in the collaborative electronic mail message and a pre-existing commitment in the electronic calendar; and if a conflict exists, then displaying an indication of the conflict.

34. One or more computer-readable media as recited in claim 27, wherein the plurality of instructions further cause the one or more processors to perform acts including: accessing an electronic calendar maintained by a task manager component; and identifying conflicts between times included in the collaborative electronic mail

messages and commitments in the electronic calendar.

35. One or more computer-readable media as recited in 34, wherein the identifying comprises identifying conflicts between commitments in the electronic calendar and collaborative electronic mail messages being authored at the computer as well as collaborative electronic mail messages received at the computer.

WEST**End of Result Set**

Generate Collection

Print

L7: Entry 19 of 19

File: USPT

Oct 21, 2003

DOCUMENT-IDENTIFIER: US 6636888 B1

TITLE: Scheduling presentation broadcasts in an integrated network environment

Abstract Text (1):

An integrated environment for scheduling a presentation broadcast that allows a user to seamlessly schedule, make changes, replace, and reschedule a presentation broadcast from within a presentation design application program. The system leverages many of the features of Microsoft's OUTLOOK.TM. program to schedule a network presentation broadcast of a presentation broadcast from within the presentation design application program that is used to create or open the presentation. The user enters information concerning the presentation broadcast while within the presentation design application program, which is then automatically inserted into a meeting request message automatically sent to a list of prospective attendees of the presentation broadcast that the user has identified. The meeting request message also contains user-entered scheduling information, which is employed to automatically schedule a presentation broadcast meeting in the electronic calendars of those message recipients who choose to attend the presentation broadcast. The system also automatically schedules the presentation broadcast in the user's electronic calendar, which provides a reminder to the user to start the presentation broadcast a predefined interval before the scheduled time. Additionally, the automated scheduling is implemented for a presentation broadcast that is to originate from an Internet web server.

Drawing Description Text (19):

FIG. 17 illustrates the relations between the lobby page and an embedded I status page used for updating the audience message;

Detailed Description Text (36):

Returning to FIG. 1, the logic next flows to a block 120, which launches a meeting request dialog 500 in the OUTLOOK.TM. program, as shown in FIG. 7. The meeting request dialog comprises a modeless window having a top-level text menu 502, and an icon menu bar 504. The meeting request dialog also includes an appointment form 506, and an attendee availability form 508. The appointment form is the first form displayed by default when meeting request dialog 500 is launched. The appointment form includes a header section with a "to . . ." recipient button 510 and an associated recipient list field 512, a subject field 514, a location field 516, an online meeting checkbox control 518, and a "meeting type" pulldown menu 520. The header section may also contain a user-notification message area 521.

Detailed Description Text (39):

Many of the fields on the appointment form will already be completed, based on the information that was entered in the presentation broadcast schedule and server options dialogs. For example, the description edit field will already be completed with information concerning the presentation broadcast that was entered in the schedule dialog, including the title, description, speaker, contact, and other information, as applicable, based on the server options previously selected. Subject field 514 will contain the presentation broadcast title, while location control 516 will already be completed with the location of the presentation broadcast as entered in the presentation broadcast schedule discussed above. Additionally, the URL property in the OUTLOOK.TM. program will be set so that the browser will automatically be launched by the OUTLOOK program if installed on any of the remote computers of the attendees. The online meeting checkbox 518 will automatically be checked, and the "meeting type" pulldown menu control will default to the meeting type selected above, e.g., it will display "NewShow Services" if a NETSHOW server was previously selected.

Detailed Description Text (40):

The user must identify who is prospectively invited to attend (view) the presentation broadcast by filling in the message recipient list field. The user may manually enter attendees into the message recipient list field, or click on the "to . . ." recipient button to bring up an address book from within OUTLOOK, from which the desired attendees can be designated. The user may enter a single recipient, multiple recipients, or choose one or more predefined lists of recipients, such as all employees of a corporation above a specific management level or in a specific workgroup. OUTLOOK also allows the user to specify required and optional attendees. The user may optionally click on the attendee availability tab to bring up a form containing schedule information pertaining to the attendee(s) that were identified in recipient list field 512. This feature is standard in OUTLOOK and need not be further discussed herein. Note that by their nature, multicast events are accessible by anybody who knows the uniform resource locator (URL) for the presentation, which contains an IP address. Uninvited attendees can thus tune in to the presentation broadcast, if they somehow learn of the event and the corresponding URL from which the presentation broadcast will be transmitted.

Detailed Description Text (67):

The URL that is used to launch the ASP page at Microsoft.com contains embedded information that a control on the ASP page deciphers so the further processing can be performed. The embedded information includes the location of the global.js file, the LCID (a language identifier, e.g., 1033 for English), and the status of the request from POWERPOINT. The form of the URL is as follows:

```
http://officeupdate.microsoft.com/office/redirect/fromOffic9/ PresBroadcasting.htm?DPC=%ProductCode%&DCC=%AppComponentCode%&AppName=%ApplicationName%&HelpLCID=%LCID%&FileServerLoc=""&Status=""
```

Detailed Description Text (69):

The values of DPC, DCC, and AppName are all Microsoft related information. The values for Status are "Schedule", "Update", "Begin", and "Delete."

Detailed Description Text (71):

Selection of a third party provider creates a URL that launches a web page corresponding to the selected third party provider, as shown by a block 142. The URL contains embedded information that is used by a control on the third party provider's web page to receive the file server location, LCID, and status. The URL will be in one of the following forms, depending on whether its corresponding web page is to be used for scheduling or broadcasting a presentation. [http://www.<3rdparty>.com/schedule.htm?HelpLCID=%LCID%&FileServerLoc="FileServerLoc"&Status="Schedule"](http://www.<3rdparty>.com/schedule.htm?HelpLCID=%LCID%&FileServerLoc='FileServerLoc'&Status='Schedule')
[http://www.<3rdparty>.com/broadcast.htm?HelpLCID=%LCID%&FileServerLoc="FileServerLoc"&Status="Begin"](http://www.<3rdparty>.com/broadcast.htm?HelpLCID=%LCID%&FileServerLoc='FileServerLoc'&Status='Begin')

Detailed Description Text (73):

In the case of scheduling a new broadcast, the Status value will be "Schedule," and the URL will target the schedule.htm page on the third party's web server. In addition to the embedded information, the global.js file is also forwarded to the targeted web page, which can download the global.js file via a control (e.g., an FTP-type control) on the page or third party web site.

Detailed Description Text (90):

At this point the browser is launched with a Status="Begin" query string so that the Microsoft.com ASP page can appropriately switch it's user interface and links to send the user to the third party provider's begin presentation broadcasting page. POWERPOINT then responds to a WINDOWS system message, 0x041E (1054 decimal), which is sent by the third party provider web page to signify that the provider is ready to begin the presentation broadcast. Upon receiving this message, POWERPOINT starts in presentation broadcast mode and begins streaming audio/video content to the third party via its internal WINDOWS Media Encoder. The third party broadcasts the HTML slides it has already uploaded in concert with the streaming audio/video content it receives in real time on the NETSHOW server to attendees who link their browsers to the URL assigned for the presentation broadcast. The streaming audio/video content contains embedded commands that indicate when individual slides should be displayed during the presentation broadcast. When the presentation broadcast is completed, a "SlideShow Ended" event is sent to the third party provider so that the provider can end the presentation broadcast.

CLAIMS:

10. The method of claim 1, further comprising the step of automatically scheduling the time and date of the presentation broadcast into an electronic calendar of the user to provide a reminder of the presentation broadcast to the user a predefined interval of time before the time it is scheduled to occur.

14. A computer-readable medium having computer-executable instructions for performing the steps of: (a) enabling a user to select a presentation to be broadcast within a presentation design application program; (b) enabling entry of presentation broadcast information for the presentation selected from within the presentation design application program; (c) enabling a time and date to be selected for the presentation broadcast; (d) enabling a list of prospective attendees to be provided for the presentation broadcast; (e) transmitting the presentation broadcast information and time and date to the prospective attendees on the list, over the computer network; and (f) automatically scheduling the time and date of the presentation broadcast into an electronic calendar of the user to provide a reminder of the presentation broadcast to the user a predefined interval of time before the presentation broadcast is scheduled to occur.

22. The system of claim 19, wherein the information management application program creates an entry into an electronic calendar of the user, specifying the time and date of the presentation broadcast.

WEST

Generate Collection

Print

L6: Entry 14 of 29

File: PGPB

Apr 18, 2002

DOCUMENT-IDENTIFIER: US 20020046076 A1

TITLE: Multi-nodal meeting planning system and method

Summary of Invention Paragraph (7):

[0008] In addition to handling group participants and event scheduling, the meeting planner must also coordinate the services and/or products being purchased through various vendors. Group events can consist of an unlimited number of individual components, such as transportation, site rental, entertainment, table linens, activities and just about any other conceivable component to operate a successful event. These services or products are typically acquired through a system of Service Requests ("SRs") or Request For Proposals "RFPs"). Generally all the details of the group program are handled by a variety of documentation and storage methods. Additionally, a majority of vendor offerings and costing/pricing information is stored in paper files. Correspondence with participants, clients (i.e., customers of a meeting planner), and vendors (i.e., sellers of goods and services consumed in a meeting or other group activity associated with a meeting) is generally accomplished using word processor documents that are mailed or faxed to the intended recipient. When correspondence is required that contains information stored in spreadsheets or other digital media, word processing mail merges or manual document generation are usually used to create these documents.

Summary of Invention Paragraph (8):

[0009] Currently, many aspects of the actions required to plan and operate a successful group meeting are accomplished in an disjointed manner through the use of individual word processors, spreadsheet programs, and contact databases, which results in a wide variety of meeting data being stored in a wide variety of formats and locations, much of which is duplicated. These methods, while they get the job done, are error-prone, provide no way of ensuring that common data is accurate from place to place, and are accessible only by particular individuals and locations. The issue becomes more significant during the actual operation of the meeting. Documentation needed to operate events and service individual participant needs is voluminous and once generated, it is difficult and time consuming to modify every copy of the documentation in use.

Summary of Invention Paragraph (10):

[0011] The large players in the group meeting industry, typically the hotel chains, airlines, and convention centers, are extremely competitive and have, in recent years, attempted to reduce pricing and begun to target the end user directly to increase profit margins and remain competitive amongst themselves. As a result, there has been continued erosion of fees and commissions historically paid to middle tier group meeting service providers. There is increasingly a need for these providers to enable themselves to provide a wider variety of services and products to their clients and to operate in a more efficient and cost-effective manner. An improved meeting planning system would also take advantage of widespread Internet access to support meeting planning system access from multiple locations, including client business offices, meeting sites, and event sites.

Summary of Invention Paragraph (11):

[0012] Although there are a variety of on-line travel and planning services available on the Internet, they are typically of limited scope and completely on-line based, i.e., an Application Service Provider ("ASP"). ASPs, such as GetThere.com, of Menlo Park, Calif., and Event411.com, of Marina del Rey, Calif., do not provide any functionality unless an Internet connection is present and address only a single aspect of the operations needed to successfully plan and operate a group meeting. ASPs do not address the need of meeting planners to have complete access and control of all aspects of the group meeting. Most of ASPs have a policy that any data entered into their systems becomes the property of the ASP service provider. There is often sensitive

information involved which corporations are reluctant to release to a third party for use in the operation of a group meeting.

Summary of Invention Paragraph (13):

[0013] The Multi-nodal Meeting Planning System and Method ("MMPS" or "PLANITDIRECT.TM. System") invention provides an innovative method for simultaneous use and updating of the data needed to plan and operate a group meeting, including comprehensive data access and standalone operation, in one or more primary locations and one or more remote locations by using a plurality of nodes in a distributed architecture, multi-node system that allows the users of the software, the client, vendors, and group participants in various locations to access and manipulate data simultaneously while assuring that the data set at a primary node remains accurate and up-to-date. Any changes and/or additions that are made at a remote node and updated on one primary node, and can then replicated to any other primary nodes. Security and confidentiality issues are addressed by the exporting only the information that the meeting planner or client deems appropriate for the operation of the group in the field environment. In addition, advantages are achieved because the group participants are able to access their individual bookings, review current reservations, modify or add to their selected activities on their own, without requiring the expense and scheduling of staff. The term "remote node" refers to any node other than a Local Node.

Summary of Invention Paragraph (17):

[0017] "Resource Browser Node" or simply "Resource Browser", which runs PLANITDIRECT.TM. application software and is accessed via any commercially available Web browser software, and contains vendor information and resources that can be employed by meeting planners, clients, and participants;

Summary of Invention Paragraph (18):

[0018] "Vendor Node", which is a node running PLANITDIRECT.TM. application software and is the access point for vendors to populate the Resource Browser with their resources, services, and products and to confirm booked reservations generated by the Remote Node and is accessed by the Local Node, a Web browser interface, or a Vendor Version. "Vendor Version" is a version of the Meeting Planning software tailored to a vendor; and

Summary of Invention Paragraph (21):

[0021] The invention also provides a method for accurate electronic booking, tracking and documentation of event and activity participation and payment. The process is centered around the features and capabilities of the PLANITDIRECT.TM. System, a destination and reservation management system which is designed to handle all aspects of group travel planning and implementation and which can interface with a remote Internet server to generate a variety of specifically targeted and accessed e-commerce and participant registration web sites for use by selected participants, clients, or resellers. One of the key features is direct payment to vendors via transaction servers operated by the PLANITDIRECT.TM. System. This allows vendors to offer events and activities at reduced pricing to selected purchasers by eliminating the mark-ups generally added by resellers, though resellers can benefit from the system as well. The intention of this invention is to provide a method for interactive sharing of information between vendors of events, activities and products and those who purchase them.

Summary of Invention Paragraph (24):

[0024] Participant names and contact information relevant to implementation of the group meeting;

Summary of Invention Paragraph (25):

[0025] Dates, times and other schedule information related to booked activities or events;

Summary of Invention Paragraph (27):

[0027] Contact information for clients, intermediaries or resellers purchasing the activities;

Summary of Invention Paragraph (32):

[0032] The PLANITDIRECT.TM. System provides a single interface to a meeting planner or person otherwise involved in group or individual travel to access and build a comprehensive itinerary of events and available activities for participants of the designated group of people traveling. The system is designed around the central concept of managing a group of people for a specified length of time at a particular

destination or destinations. The system allows the user to build groups and events by utilizing a concept called "Resources". These resources can be anything from a 8 hour private charter fishing boat to filigree doily sets for a table setting of 2000 to a 50 passenger minibus used to transport a group of people. These resources can be loaded in by hand, imported via "data sets" or obtained through the net via the Resource Browser. The user can use the system to peruse the resources available in a particular destination and then choose to store selected resource for local use and availability to all users or just utilize that resource for a particular group. When the resource is transferred to the Local Node, it can then be adjusted, modified and customized to suit the needs of the particular planner or group meeting requirements.

Detail Description Paragraph (2):

[0039] As shown in FIG. 1, the Local Node (110) consists of the PLANITDIRECT.TM. System Meeting Planning software that allows the planner who is creating or coordinating a group program to have access to all the pieces required to assemble a successful group program. This software interfaces with all the other aspects of the PLANITDIRECT.TM. System and serves as a cohesive location where decisions can be made, reports can be generated, details can be tracked and financial information can be evaluated. Communications between the various nodes of the system are accomplished in a variety of ways. Most routine operations between the Remote Field nodes are accomplished by using encrypted and signed e-mail messages through a dedicated e-mail server integrated into the Remote Node (150) which process the requests submitted to it via those messages and returns either a set of data that was requested or a confirmation that a specific instruction was successfully implemented. For instance, if a Remote Field Node (120) sends a request for all new participant records, the Remote Node responds with an encrypted, signed message containing the requested data back to the originating Remote Field Node (120). If a Remote Field Node sends a request that does not require data return, such as an instruction to mark a particular record as having been read by that particular Remote Field Node, the Remote Node sends back a message that simply states that the instruction has been successfully implemented.

Detail Description Paragraph (8):

[0045] "Resource Browser Node" (130) or simply "Resource Browser", which runs PLANITDIRECT.TM. application software and is accessed via any commercially available Web browser software, and contains vendor information and resources that can be employed by meeting planners, clients, and participants;

Detail Description Paragraph (9):

[0046] "Vendor Node" (140), which is a node running PLANITDIRECT.TM. application software and is the access point for vendors to populate the Resource Browser with their resources, services, and products and to confirm booked reservations generated by the Remote Node and is accessed by the Local Node, a Web browser interface, or a Vendor Version. "Vendor Version" is a version of the Meeting Planning software tailored to a vendor; and

Detail Description Paragraph (17):

[0054] As shown in FIG. 3, the Local Node (310) contains all of the information required for a group in a central data repository that contains all of the data for all groups being handled by the meeting planner. This includes, but is not limited to: Client, Contact, Association and Employee Data; Group Start/End Dates, Group Location, Default Airport, Default Hotel, Activities, Group Events, Transportation Information, Participant Travel Data, Staffing Schedules, Main Resources, Group Resources and all other data needing to be tracked, manipulated or accessed for the successful operation of a group.

Detail Description Paragraph (18):

[0055] The Local Node (310) is typically located on an individual PC, a secure LAN or a company LAN and is not accessible to the general population. The local administrator of the system upon which the Local Node is located defines access control. Individuals designated as having permission to access the compiled information may import individual sets of data into the Local Node from a variety of sources to accommodate the cohesive access of all group data. The Local Node replicates individual group meeting data sets as well as all related information to a particular group meeting for use in the field, typically on laptop computers via a LAN. These replicated data sets then become Remote Field Nodes (320). Additionally, the Local Node controls what information is to be sent or made available to the Remote Node based upon user settings. The Local Node updates the main data set via a polling system or real-time updates depending upon the type of remote node implemented for a particular group. These methods are described in the above section, Shared Remote Data Real-Time Update

System.

Detail Description Paragraph (21):

[0058] The invention described herein consists of a Remote Field Node export and re-assimilation process. This allows the use of PLANITDIRECT.TM. Meeting Planning software designed to be in operation at the actual group event location and immediate access to cost/price reports and logistics documentation needed for the handling of group event happenings.

Detail Description Paragraph (23):

[0060] The information sent out encompasses all data previously entered and needed to coordinate the logistics of handling a particular group of Participants. This may include transportation movements, individual activity bookings, event scheduling, staff scheduling, participant accommodations, group resources and client communications histories as well as any other information necessary for the successful operations of a group.

Detail Description Paragraph (27):

[0064] Transfer of data to the blank data set container in a particular order to assure that relationships required by relational database techniques are adhered to. For example: The main group cannot be created before the client contact information because the group is dependent on the contact ID number to maintain data integrity and to identify which client a particular group belongs to;

Detail Description Paragraph (32):

[0069] Export of user information and security access levels;

Detail Description Paragraph (33):

[0070] Export of all documents contained in the folder associated with the group for access in the field.

Detail Description Paragraph (34):

[0071] Remote Field Data Set access at the group location;

Detail Description Paragraph (39):

[0076] The PLANITDIRECT.TM. System Meeting Planning software allows access to designated local storage media at the click of a button. Contact records, Employee, Association and Group records can be associated with a particular folder located on a local hard drive or through a local network.

Detail Description Paragraph (40):

[0077] The association of particular pieces of data with a locally accessible file storage device allows for the automatic saving of documents in a particular common location when they are generated by the software. These documents may consist of word processing documents, flat-file database files, and static report "snapshots". The selected folders are then easily accessible from within the software at the click of a button.

Detail Description Paragraph (42):

[0079] The PLANITDIRECT.TM. System Meeting Planner software provides a common repository for the storage of contact information for Clients, Vendors, Suppliers, Employees, and Associations that can be used repeatedly to generate group participant lists as well as a comprehensive contact history record.

Detail Description Paragraph (44):

[0081] About Company-Vendor Data Details Availability.

Detail Description Paragraph (45):

[0082] The PLANITDIRECT.TM. System Meeting Planner software provides a method of accessing detailed vendor and product information electronically without have to physically locate the information needed.

Detail Description Paragraph (46):

[0083] The software stores vendor company and detail information in a format that is accessible from a variety of places in the group program areas as well as the contact management areas of the software. A small text/graphics box is displayed in these areas that, when double-clicked, launches a detailed company information document in a popup viewer. This document may have services provided, contact information, pricing information, as well as a means to obtain additional information about this company

(such as a World Wide Web link).

Detail Description Paragraph (50):

[0087] In addition, this departure information can be accessed via the Live Meeting Web Site located on the Remote Node which is accessible to the participant via a Web Browser.

Detail Description Paragraph (61):

[0098] Both you and your luggage are scheduled to depart the Hotel at 4:24 p.m. from the Four Season's group entrance.

Detail Description Paragraph (68):

[0105] The PLANITDIRECT.TM. System Meeting Planning software provides a single repository of schedule information for each group participant. Comprehensive itineraries can be generated which include all schedule related information pertaining to a particular group participant.

Detail Description Paragraph (69):

[0106] Because all schedule related information pertaining to individual group participants is available to the user, it is a straightforward process to generate comprehensive itineraries for group participants which include everything from arrival flight information, activity schedules, events which to which the participant has subscribed, hotel departure information and departure flight information. This personalized participant itinerary can then be distributed to each participant via printed page or through electronic media.

Detail Description Paragraph (70):

[0107] In addition, this itinerary can be accessed via the Live Meeting Web Site located on the Remote Node which is accessible to the participant via a Web Browser.

Detail Description Paragraph (72):

[0109] One of the key services provided by a meeting planner is the creation of a custom event for a group. These events typically consist of a multitude of components such as transportation, activities, catering, entertainment and a variety of others. Usually these events are paid for by the group client at either a bulk sell price or a fixed mark-up of each component of the event and all group participants have free access to these events. Occasionally, the event will be designed to have the actual participants pay for their entry into a particular event. Currently, there is not a method for easily accomplishing this.

Detail Description Paragraph (109):

[0146] Service Requests or Requests For Proposals are generated by the group meeting planner to order products or services necessary for the successful operation of a group program. These also serve to check for the availability of a particular service on a particular day and time.

Detail Description Paragraph (148):

[0185] Additional details for the web site such as access settings, whether a single password will be used to access or whether individual passwords for participants will be required, whether participants will be able to change their arrival/departure information, whether participants will be able to enter their own personal entries for an itinerary, whether higher level access is allowed for administration purposes, etc.;

Detail Description Paragraph (149):

[0186] Which participant names and details are to be sent to the web site? For instance, a user may choose to only allow primary participants to have access and not guests or may want to have all participants to have access to the web site; and

Detail Description Paragraph (153):

[0190] The Web Builder also determines, at the discretion of the user, which menus and pages are available for each. Participant logged into the Live Meeting Web Site. These can be selected individually or based on common criteria by the planner using the PLANITDIRECT.TM. System Meeting Planning software and allow for the ability to allow access to particular pages and areas of the Live Meeting Web Site. This allows the planner to set up particular pages which are oriented towards staff, for instance.

Detail Description Paragraph (188):

[0225] Detailed scheduling information including start and end times with descriptions;

Detail Description Paragraph (193):

[0230] Staff scheduling; and

Detail Description Paragraph (201):

[0238] Service request information including whom to send the request to, any schedule information required and any inclusions associated with the component.

Detail Description Paragraph (203):

[0240] The Remote Field Node is generated by the PLANITDIRECT.TM. System Meeting Planner software and duplicates all of its capabilities, but is designed to be more portable, contains only the data necessary to operate a single group meeting and addresses the needs of actually implementing the operations of a group program at its location. This aspect of the system allows the personnel actually operating a group program in the field to access and modify all aspects of the group program. A group program is not limited to a single Remote Field Node for a group meeting program. As many copies as necessary may be generated, each with its own identifier for distinct communications with the Remote Node. The Remote Field Node software interfaces with both the Remote Node and the Meeting Planning software to allow all aspects of the group operations to be kept up to date by every portion of the PLANITDIRECT.TM. System. This includes the remote group planner, the vendors involved and other Remote Node software users for a particular group. The software is designed to be stand-alone and secure and is not dependent upon an electronic connection to operate. This allows for the use of the software in sometimes remote and unconnected situations. If no connectivity is available during the operation of the group meeting, the data collected in the Remote Node data set can be re-integrated back into the PLANITDIRECT.TM. System Meeting Planning software via LAN or VPN connection.

Detail Description Paragraph (206):

[0243] As shown in FIG. 6, the Remote Node is the nexus of communications between the various other nodes of the system as well as being the host for the Live Meeting Web Site. This node takes the information compiled by the group program planner and builds a group specific web site which can then be interacted with by the individual participants for activity purchase, itinerary, scheduling and communications functions.

Detail Description Paragraph (207):

[0244] The Remote Node consists of the data determined by the PLANITDIRECT.TM. System Meeting Planning software to be necessary for access during the operation of the group meeting program. Remote data is accessible from any computer or digital device via the World Wide Web, Wireless Application Technology or other electronic means. The concept allows for the use of technologies not yet invented or implemented for instant data access from locations removed from the home office and not requiring a dedicated device for access.

Detail Description Paragraph (208):

[0245] Security of the remote data typically consists of user names and passwords to determine the level of access allowed. For instance, group participants may be limited to being able to book activities, view their personal itineraries, modify their arrival and/or departure information or access files designated as public group documents. Individuals logging in as operators would be able to access all of the above information as well as accessing Event, Transportation and Staffing details and Documents designated as operator accessible. Individuals logged in as administrators may be able to access all of the above and also have the ability to modify staffing schedules, change or add available activities, post information intended to be viewed by all participants and modify pricing information as well as accessing all documents available to the Remote Node.

Detail Description Paragraph (211):

[0248] The Remote Node also handles all transactions generated by the Live Meeting Web Site. This includes processing credit cards, in the case of financial transactions, as well as processing reservations for transaction-less bookings such as occurs when a participant reserves a spot on a "Hosted" event, i.e. costs covered by the Event organizer, with no charge to the participant. All reservations are also forwarded to the Vendor Node for access by vendors enrolled in the PLANITDIRECT.TM. System as determined by settings selected by the user in the PLANITDIRECT.TM. System Meeting Planning software. In the event that the user has elected to have confirmations functions performed by the vendors themselves, the Remote Node retrieves confirmations

from the PLANITDIRECT.TM. Vendor Node and passes this information on to all the various nodes, both Local and Remote, that are affected by the confirmation.

Detail Description Paragraph (213):

[0250] As shown in FIG. 4, the Resource Browser Node (430) is the repository for all universal resources available to users of the PLANITDIRECT.TM. System. The Resource Browser (430) in the software consists of a common pooling of resources used in a group. The Resource Browser consists of many different types of data, all sharing a common storage format. This allows the resources to be used in a variety of places in the group meeting planning software without having to duplicate individual resources for multiple use. These resources are broken up into several categories: Activities, Transportation, Event Components, Golf Components, Food & Beverage, Labor, Meeting Components and Accommodations. These resources contain vendor information, resource description, pricing information as well as details particular to the resource listed. For instance: Activity type resources also contain fields for input of special instructions and check-in locations to be used when generating vouchers as well as brochure descriptions used when generating documents or sending to remote locations for electronic access. The Resource Browser (430) provides a completely independent set of resources for an individual group allowing the customization of the resources for that group to occur without affecting the original resources or resources set for other group programs. The resources actually used in a particular group meeting are duplicates of resources in the Resource Browser Node. Resources are located by using a web interface and browsing through the available resources for a particular destination. A series of filters and parameters can be used to narrow down the search, such as price point, resource type, availability, etc. Resources may be pre-selected on the Resource Browser web site or may be imported into the PLANITDIRECT.TM. System Meeting Planning software real-time. Storage of selected sets of resources on the Resource Browser web site can be segregated by group for later decision and import. This allows the meeting planner to access and assemble these resources with the use of a web browser from anywhere without having to have a copy of the PLANITDIRECT.TM. System Meeting Planning software, whether local or remote, in front of him or her to do so.

Detail Description Paragraph (216):

[0253] PLANITDIRECT.TM. Vendor Node: This node interfaces with the Resource Browser to allow vendors enrolled in the system to enter their product descriptions, details, pricing and availabilities to allow meeting planner users of the system to access them. The software allows the vendor or a third party to access, maintain, add or modify the wares or services being provided by the vendor, which is being offered to the users of the PLANITDIRECT System. This interfaces directly with the destination management software and becomes the "resources" available by that vendor for a particular destination. This can be accomplished directly via a web site or can be compiled with standalone software and uploaded to the PLANITDIRECT.TM. System; and

Detail Description Paragraph (227):

[0264] Information regarding the provider of the resource including contact information, cancellation policies, buyout policies, insurance requirements, etc.;

Detail Description Paragraph (231):

[0268] As shown in FIG. 5, the PLANITDIRECT.TM. Vendor Node (540) allows the vendor to interface directly with the PLANITDIRECT.TM. System and receive bookings and purchases real-time, as well as enter vendor resource information to made available to users of the PLANITDIRECT.TM. System. This can be done via a web site interface or via a flavor of the PLANITDIRECT.TM. Vendor software. The software can also be configured to retrieve and upload data at specific intervals for those vendors without always on connections. In a worst-case scenario, the software can be used as a stand-alone reservations system with manual input by the vendor. The software also enables the vendor to track his wares, and allows for the generation of internal schedules, confirmations and purchaser details. The system is also capable of handling the complexities of activity booking and vendor resource management. In its simplest form, it consists of a standalone PC based reservation system. In its more robust use it has the capability of handling bookings, sales and confirmation via direct electronic communication.

Detail Description Paragraph (234):

[0271] The specific details pertaining to each vendor are entered and stored for use by individuals or other processes accessing this information. This can be as simple as mailing address and contact phone number data and can be as comprehensive as complete contact lists defining departments, payment method information, physical location

information, electronic contact information (e-mail, web-site, etc.), deposit requirement policies, cancellation policies and vendor insurance information as well as any other information pertaining to the vendor which may be germane in assisting a prospective purchaser in selecting the particular vendor to supply whatever product may be needed.

CLAIMS:

1. A meeting planning system using a distributed architecture containing a plurality of nodes, comprising: a meeting planning application program running on a server node for entry of data, browsing resources, planning activities and scheduling for a meeting event, generating at least one meeting plan, and exporting the at least one meeting plan with selected meeting data to at least one Remote Node selected from the group comprising: a World Wide Web server node that is accessible to meeting attendees, a secure server node operated by a vendor, a World Wide Web server node operated by a vendor, a secure server node operated by a client, and a World Wide Web server node operated by a client.

2. A system for planning and managing transportation, accommodation, activity, and event reservations using a distributed architecture containing a plurality of nodes, comprising: a means for accessing information on location-specific vendor resources and for handling direct bookings by members of a group related to the members' participation in a meeting; a means for vendors to receive reservations and to transact purchases in real-time; a means for creating and updating at least one World Wide Web site that provides meeting information, entry and updates of reservations, transactions with vendors, and posting of the updated information to selected nodes within the distributed architecture; a central management system and destination router for routing meeting planner inquiries to the appropriate node; and a transaction server node that handles at least one transaction selected from the group comprising: billing, commission, and payment.

WEST

Generate Collection

Print

L6: Entry 15 of 29

File: PGPB

Feb 7, 2002

DOCUMENT-IDENTIFIER: US 20020016734 A1

TITLE: Integrated household management system and method

Summary of Invention Paragraph (4):

[0003] A number of applications for organizing data and schedules are known in the art. These applications can be either stand-alone or can operate among groups of users which may be linked, for example, via an Intranet. Also known in the art are Internet-based services that provide promotional and advertising material, such as special sales, special fares or merchandise coupons, to users who may have signed up for these services or may be targeted by advertisers and retailers based on demographic user profiles. Users may also select specific areas of interest, but these services may be limited to selected topics, such as travel, cars, etc. In daily family life, however, various tasks which may be interrelated have to be performed, some of them repeatedly and at regular time intervals. Family events, such as birthdays, anniversaries, have to be monitored; there may be an interest in consolidating financial statements and shop for the best available offers based on the family resources.

Summary of Invention Paragraph (7):

[0005] The invention is directed to a method for managing household activities of a user connected to a service system, in particular via the Internet. According to the method, the user provides to the service system user data which include demographic user information and interactive user behavior characteristics, whereafter the service system--in response to the user data--provides the user with a household management tool. The user schedules the household activities by using the household management tool, wherein information about the scheduled household activities is transmitted from the user to the service system. The service system associates a scheduled household activity with an incentive and transmits the incentive to the user.

Summary of Invention Paragraph (8):

[0006] Embodiments of the invention may include one or more of the following features. The incentive may be provided to the service system by a provider, such as a retailer, a manufacturer, a service provider and/or a clearing house. The household management tool may include a scheduler and/or calendar and the scheduled household activity may be a meal schedule relating to recipes managed by the service system.

Detail Description Paragraph (3):

[0017] FIG. 1 depicts one embodiment of a system 10 according to the invention for managing family and household activities of a subscriber. Specifically, FIG. 1 illustrates a system 10 wherein a plurality of subscriber systems 12 connect through a network 20 to the server 14. The server 14 connects to a proprietary database 16 maintained by the server 14 and similarly connects, optionally by direct secure lines, to a plurality of service providers or merchandise vendors 18. The elements of the system 10 can include commercially available systems that have been arranged and modified to act as a system according to the invention, which allows a subscriber to carry out integrated household activities, and optionally generate records of these integrated activities. The system 10 of FIG. 1 employs the Internet to allow a subscriber at a remote client, the subscriber systems 12, to access a central server, the depicted central server 14, to login to an account maintained by that server, and to employ the services provided to that account to manage family and household activities of the subscriber.

Detail Description Paragraph (6):

[0020] For the depicted system 10, the client systems 12 can be any suitable computer system such as a PC workstation, a Web-TV, a monitor or display installed, for example, in a kitchen area of the subscriber, a wireless communication device, or any other such device, equipped with a network client capable of accessing a network server and

interacting with the server to exchange information with the server. In one embodiment, the network client is a web client, such as a web browser that can include the Netscape web browser, the Microsoft Internet Explorer web browser, the Lynx web browser, or a proprietary web browser, or web client that allows the user to exchange data with a web server, and ftp server, a gopher server, or some other type of network server. As mentioned above, the client 12 and the server 14 rely on an unsecured communication path, such as the Internet, for accessing services on the remote server 18. To add security to such a communication path, the client and the server can employ a security system, such as any of the conventional security systems that have been developed to provide to the remote user a secured channel for transmitting data over the Internet. One such system is the Netscape secured socket layer (SSL) security mechanism that provides to a remote user a trusted path between a conventional web browser program and a web server. Therefore, the client systems 12, the vendor systems 18 and the server system 14 may have built in 128 bit or 40 bit SSL capability for establishing an SSL communication channel between the clients 12 and the server 14. Other security systems can be employed, such as those described in Bruce Schneir, Applied Cryptography (Addison-Wesley 1996). Alternatively, the systems may employ, at least in part, secure communication paths for transferring information between the server and the client. For purpose of illustration however, the systems described herein, including the system 10 depicted in FIG. 1 will be understood to employ a public channel, such as an Internet connection through an ISP or any suitable connection, to connect the subscriber systems 12 and the server 14.

Detail Description Paragraph (7):

[0021] The server 14 may be supported by a commercially available server platform such as a Sun Sparc.TM. system running a version of the Unix operating system and running a server capable of connecting with, or exchanging data with, one of the subscriber systems 12. In the embodiment of FIG. 1, the server 14 includes a web server, such as the Apache web server or any suitable web server. The web server component of the server 14 acts to listen for requests from subscriber systems 12, and to in response to such a request, resolves the request to identify a filename, script, dynamically generated data that can be associated with that request and to return the identified data to the requesting subscriber system 12. The operation of the web server component of server 14 can be understood more fully from Laurie et al., Apache: The Definitive Guide, O'Reilly Press (1997). The server 14 may also include components that extend its operation to accomplish the integrated household activities described herein, and the architecture of the server 14 may vary according to the application. For example, the web server may have built in extensions, typically referred to as modules, to allow the server 14 to perform operations that facilitate the integrated household activities desired by a subscriber, or the web server may have access to a directory of executable files, each of which files may be employed for performing the operations, or parts of the operations, that implement the integrated household activities of the subscriber. Thus it will be understood that the server 14 may act as a household activities transaction server according to the invention that configures the work station hardware supporting the server 14 to act as a system according to the invention.

Detail Description Paragraph (8):

[0022] The server 14 may couple to a database 16 that stores information representative of a subscriber's account, including information regarding the subscribers accounts, including passwords, user accounts, user privileges and similar information. The depicted database 16 may comprise any suitable database system, including the commercially available Microsoft.RTM. Access.TM. database, and can be a local or distributed database system. The design and development of database systems suitable for use with the system 10, follow from principles known in the art, including those described in McGovern et al., A Guide To Sybase and SQL Server, Addison-Wesley (1993). The database 16 can be supported by any suitable persistent data memory, such as a hard disk drive, RAID system, tape drive system, floppy diskette, or any other suitable system. The system 10 depicted in FIG. 1 includes a database device 16 that is separate from the server station platform 14, however, it will be understood by those of ordinary skill in the art that in other embodiments the database device 16 can be integrated into the server 14.

Detail Description Paragraph (9):

[0023] FIG. 2 provides a functional block diagram of one server 14 for managing family and household activities of the subscriber and further depicts the data flow diagram of one example of a subscriber's use of the server 14 to perform an integrated financial transaction between the financial service providers 28 and 30. Specifically, FIG. 2 depicts a data flow diagram wherein a subscriber 12 employs a user interface 32, such as an HTML page, to provide user input to the server 14. As can be seen from FIG. 2,

the server 14 manages the subscriber's account and also manages communication between the subscriber and the various companies/retailers/clearing houses in which the subscriber either has a direct interest (response by clicking through) and/or which the server administrator deems to be of interest to the subscriber based on a profile of the subscriber's integrated household activities, which can include, for example, demographic user information and interactive user behavior characteristics. Specifically, the server 14 is described as a functional block diagram that includes, among others, a web server 40 and a common gateway interface (cgi) module 42, 44 which can communicate web page content to and from a database 16 or a database of retailer 18. The web server 40 can be any suitable web server, for example, an Apache.TM. web server having access to a set of executable files stored in a directory accessible to the web server 14, such as the cgi-bin directory in module 42.

Detail Description Paragraph (12):

[0026] The depicted database 16 can be any suitable database system, including the commercially available Microsoft.RTM. Access.TM. database, and can be a local or distributed database system. The design and development of suitable database systems are described in McGovern et al., A Guide To Sybase and SQL Server, Addison-Wesley (1993). The database 16 can be supported by any suitable persistent data memory, such as a hard disk drive, RAID system, tape drive system, floppy diskette, or any other suitable system. The system depicted in FIG. 1 includes a database device 16 that is separate from the server platform 14, however, it will be understood by those of ordinary skill in the art that in other embodiments the database device 16 can be integrated into the system 14.

Detail Description Paragraph (50):

[0064] Contact information (address book, important dates, children names and birthdays, etc.) of friends, service companies, and the like may also be supplied and stored in the database 16 of the household management system. This contact information will be integrated with other section of the Application, making it easy for users to add to the contact information from a Home Maintenance section (e.g. add appliance retailer and/or service provider). Similarly, important dates (e.g. birthdays, anniversaries) added to a Contact Record can be automatically added to the calendar.

Detail Description Paragraph (51):

[0065] A large portion of family income is typically spent on food and services. Careful planning and taking advantage of coupon savings and other promotions may significantly reduce the expenses for such items. Accordingly, the user will be able to schedule a meal and/or other functions, step 35, as follows:

Detail Description Paragraph (54):

[0068] Building, saving and editing meals and assigning them to a schedule (e.g. Favorites)

Detail Description Paragraph (57):

[0071] The meals may preferably be integrated with a shopping list. Meals can be built and edited on a per day basis. The HTML page 32 may include multiple frames to facilitate adding to a meal. Ingredients, recipes, saved meals, generic items and branded products can all be added to a meal. Users may have the option of viewing the meal schedule on a weekly or monthly basis. Basic meals for each user can be created, such as orange juice, toast, and cereal for breakfast, ham and cheese sandwich and an apple for lunch, etc., as a starting point for creating daily meal plans. Additional stock meals may be suggested by the household management system. Stock meal categories may be arranged by region, "quick" "family style", etc. A printable view may be available. Users may have the option of previewing the applicable recipes and selecting for printing.

Detail Description Paragraph (58):

[0072] Scheduling and planning, step 35, also gives the user the opportunity to view stock recipes managed by the household management system and his own recipes. The user, however, will only be able to edit or delete his own recipes. A memo field may be provided for users to write and save notes related to other recipes. Recipes may be searched based on food group, region, nutrients, course, cookbook, preparation time, or keyword and may be assigned to a meal.

Detail Description Paragraph (60):

[0074] Another function provided by the Scheduler/Planner 35 is the creation of a shopping list. Referring now also to FIG. 4, the Application creates a shopping list, step 42, based on the meals created with the meal schedule and recipes described above

with reference to step 35. The user may add items to the shopping list from a "Quick List", available meal plans, other suggested meals, recipes, product lists, generic lists, or by searching, for example, the Internet, and/or by free-typing. The user may also integrate planned meal schedules and/or promotional offers/coupons, possibly based on product category.

Detail Description Paragraph (64):

[0078] The user may decide to plan a meal or any other purchase based on savings available from different companies or vendors. The shopping list is therefore advantageously integrated with the saving offers as follows: a button may be placed at the top of the list to display all offers relating to items on the list (including competing items, in the case of brands). An icon may indicate the availability of an offer for each item in a product class with an available offer. Clicking this icon will display any offers relating to the product.

Detail Description Paragraph (67):

[0081] Referring back to FIG. 3, the user may update his/her personal records, as described in the specification, step 36, in which case the Application 30 return to step 34. Updating may be performed anytime during the session with the household management system and/or off-line. A registered user can log on directly, preferably after entering a password, step 38, and view, for example, the scheduling/planning HTML page, step 35.

Detail Description Paragraph (68):

[0082] The user may not only rely on a computer or Web-TV as interactive devices for communicating with the household management system. For example, passive displays or displays having limited interactivity (such as displays combined with a keypad or with voice-activated controls) may be located in areas, such as the garage of the household, the kitchen and/or the laundry room to display messages alerting the user to promotional offers or scheduled service, for example, for the car or an appliance. Such devices may also be connected to wireless communication devices to provide increased flexibility in scheduling.

Detail Description Paragraph (82):

[0095] UserHome.asp--Welcomes user who has successfully logged in, allows access to member functions

Detail Description Paragraph (92):

[0103] Contact.asp--Allows a user to enter or modify contact information, uses sp_ReadContact for update ValidateContact.asp--Confirms information, performs appropriate database operation, using sp_InsertContact or sp_UpdateContact

Detail Description Paragraph (100):

Meal Schedule (see FIG. 8)

Detail Description Paragraph (101):

[0111] MealSchedule.asp--Displays monthly meal schedule

Detail Description Paragraph (102):

[0112] MealScheduleMeals.asp--Allows a user to add or remove a meal from the schedule

Detail Description Paragraph (119):

[0126] Referring now to FIG. 12, the server 14 may include a number of administration screens which may also be implemented as ASP files in a password-protected directory on the server, for example, in the server database 16. These screens will allow the household management system to view or administer data for which they are responsible, including data updates of the content, recipes (with images and thumbnails), Quick List seed items, ingredients, products, generic products, product classes, categories, departments, offer types, meals and types (e.g. "family meals", "Mexican meals", etc.), contact categories, event categories, suggested events, To-do list types (e.g. "Spring cleaning"). Where appropriate, screens could be located where Business Partners could have access to their own information, such as promotions and special offers.

Detail Description Paragraph (149):

[0155] Recurring event schedule--Store instructions on how an event may repeat, including whether daily, weekly, monthly or yearly, days of the week on which it is held, as well as a week or month cycle (e.g. 4.sup.th Thursday in November).

Detail Description Paragraph (168):

[0174] Meal schedules--Stores a subset of recurring event schedules to be used for meal patterns.

Detail Description Paragraph (180):

[0186] Referring now to FIGS. 13-20 and as described above, the invention provides a home management system and a method for using such system which organizes and manages in an integrated fashion the household activities. As seen in FIG. 13, the user 13 can review and manage in an integrated fashion various household activities, such as planning meals 132 and shopping for staples and other items 134, keeping family records 136 and pet history 135, schedule and be reminded of upcoming events 137 and organize his/her contacts 138. All these activities are integrated and interrelated, as shown with particularity in FIG. 20, which shows in detail the functional blocks of FIG. 13.

Detail Description Paragraph (181):

[0187] The different sections indicated in FIG. 20 reference each of the respective FIGS. 14-19 to which they correspond. For example, the contact information 138 of FIG. 14 may include personal data 142 of the contacts, such as names, phone numbers, addresses 144, gift ideas 148 for special occasions, as well as the names, addresses, etc., of professionals, such as home maintenance services 146. Events 137 depicted in FIG. 15 may include one-time events 132 and recurring events 154 as well as medical/health information 146 regarding doctor visits and drugs, and a calendar 148. It will be understood that the calendar 148 may be interfacing with many of the other activities, such as scheduling of meals, as indicated by the arrows in FIG. 20.

Detail Description Paragraph (183):

[0189] An important feature of the integrated household management system is the meal planner 132 which is linked to a collection of recipes 174 geared towards stock meals 172 provided by the service system or personalized meals 172 entered and/or modified by the user. The meals 132 can be scheduled, as mentioned above, with the help of the calendar 148. The meals 132 can also be linked to a shopping list 176 based on the recipes 174 which are used to prepare the meals 132. As will be apparent from FIGS. 18 and 19, the shopping list 176 can be linked with promotions, store information and the like. The shopping list can match both ingredients, like those used in recipes, as well as branded items, like Gerber and Ivory. Since there may be different products, such as Ivory Dishwashing Liquid or Ivory Laundry Flakes, intelligence should be built into the system to quickly provide context. Also, ingredients should be anticipated so that the ingredients can be quickly "matched" and "attached" to products in the shopping list.

Detail Description Paragraph (186):

[0192] The system and method described above may be integrated with systems operated by companies, retailers and clearing houses to communicate to the user promotional offers suggesting purchases, activities, services and meal schedules. Such offers can be made in the form of announcements (email, banners, windows), with explicit coupon offers that can be redeemed in a store, or by directly crediting a promotional award to a purchase made in the store.

CLAIMS:

1. A method for managing household activities of a user connected to a service system, comprising: the user providing to the service system user data which include demographic user information and interactive user behavior characteristics; the service system in response to the user data providing the user with a household management tool, the user scheduling the household activities by using the household management tool, wherein information about the scheduled household activities is transmitted from the user to the service system, the service system associating a scheduled household activity with an incentive and transmitting the incentive to the user.
3. The method according to claim 1, wherein the household management tool is a scheduler or calendar.
6. The method according to claim 1, wherein the scheduled household activity is a meal or meal schedule.

WEST

Generate Collection

Print

L6: Entry 18 of 29

File: USPT

Mar 18, 2003

DOCUMENT-IDENTIFIER: US 6535892 B1

TITLE: System and methods for exchanging messages between a client and a server for synchronizing datasets

Parent Case Text (3):

The present application is also related to the following commonly-owned U.S. patent applications, the disclosures of which are hereby incorporated by reference in their entirety, including any appendices or attachments thereof, for all purposes: Ser. No. 09/208,815, filed Dec. 8, 1998 and entitled System and Methods for Robust Synchronization of Datasets; serial No. 60/109,983, filed Nov. 25, 1998 and entitled System and Methods for Transaction-based Synchronization; serial No. 60/106,189, filed Oct. 28, 1998 and entitled System and Method for Transaction-based Synchronization; Ser. No. 09/136,215, filed Aug. 18, 1998 and entitled System and Methods for Synchronizing Two or More Datasets, now U.S. Pat. No. 6,295,541; Ser. No. 09/136,212, filed Aug. 18, 1998 and entitled Data Processing Environment with Methods Providing Contemporaneous Synchronization of Two or More Clients now U.S. Pat. No. 6,275,831; Ser. No. 08/923,612, filed Sep. 4, 1997 and entitled System and Methods for Synchronizing Information Among Disparate Datasets; and Ser. No. 08/693,677, filed Aug. 12, 1996 and entitled Scheduling System with Methods for Peer-to-Peer Scheduling of Remote User now U.S. Pat. No. 6,016,478.

Brief Summary Text (5):

With each passing day, there is ever increasing need for synchronization solutions for connected information devices. Here, information devices include for example general- or special-purpose computers of all types and sizes, Internet or intranet access devices, cellular phones, pagers, and other hand-held devices including, for example, REX PRO.TM., PalmPilot and Microsoft "Windows CE" devices, and the like.

Brief Summary Text (13):

In addition to high latency, communication environments (e.g., wireless, message-based, or Internet-based environments) may also have other characteristics such as low or intermittent reliability and availability, low or expensive bandwidth, inability to guarantee FIFO (first-in-first-out) delivery order, and the like. Each of these other characteristics introduces additional problems to the synchronization task. Furthermore, combinations of such communication characteristics makes the problems especially difficult to solve. To take just one example, consider the task of synchronizing over a communication medium that cannot guarantee FIFO delivery order for transmissions and is susceptible to high latency (e.g., latency longer than an hour or longer than a day). Suppose that, in a first synchronization, some synchronization-related messages containing data are sent across this communication medium. Suppose that the user thereafter makes changes to one of the datasets involved, in the ordinary course of using the dataset. Then, perhaps hours or days later, the user attempts a second synchronization over the communication medium. The second synchronization finishes and leaves correctly-synchronized datasets. However, the now-obsolete synchronization-related messages sent during the earlier first synchronization may now arrive at one or more of the datasets (i.e., in non-FIFO order). If these messages are obeyed, the already-correctly-synchronized datasets may become corrupted with obsolete data from the now-obsolete messages. In short, the user's desired information is endangered by a characteristic of the communication medium that affects synchronization.

Detailed Description Text (11):

The synchronizer 200 includes a synchronizer core 240, an optional User Interface 245 (UI), and client accessors including for example a first client's accessor 250 and an N-th client's accessor 253. The synchronizer core includes a synchronization engine 254 and a reference dataset 255. Each client accessor includes sufficient knowledge (e.g.,

client-specific knowledge) to enable the accessor to access (for example, read and write) information on a client's dataset and to communication such information to and from the synchronizer core 240, via the communication mediums 209. Each client accessor may and preferably does run on a same machine as the client, e.g., on a remote machine with respect to the synchronizer core. Information stored by a client accessor is preferably stored on the accessor's local machine for efficiency, or may be stored with the synchronizer core and accessed by the accessor via its connection with the synchronizer core. The Synchronization engine 254 controls the reference dataset 255, which is also referred to as the synchronizer dataset or GUD ("Grand Unification Dataset"). The GUD is for storing a super-set of data from all datasets. Together, the synchronizer core 240 and the client accessors manage the synchronization process. The optional UI 245 provides optional interactive input and output to a user during the synchronization process. The UI optionally includes a browser or terminal or similar user interface technology and enables the user to view or modify the information in the GUD to thereby provide PIM functionality using the GUD. The synchronizer of FIG. 2 may be constructed from Starfish synchronization system(s) that are described, for example, in the incorporated U.S. patent applications having Ser. No. 09/136,215, now U.S. Pat. No. 6,295,541, and Ser. No. 09/208,815 by adding the additional features and synchronization methods described in the present document.

Detailed Description Text (12):

In the remainder of this document, the terms "client" or "client dataset" alone may be used to refer to the synchronizer's client dataset accessor, and the terms "synchronizer" or "server" alone may be used to refer to the synchronizer core (e.g., core logic and/or reference dataset), for simplicity. Context should make clear the intended meaning where a distinction is actually necessary.

Detailed Description Text (18):

In the above-described approach, the serial communications connection may be made across a wireline connection or a short infrared link. With such a connection, the synchronization logic (e.g., conducted by a PC-based program) can directly access both datasets similarly to the way a program might access files in file systems. Under such an approach to dataset access, the synchronization logic may conduct communication exchanges over the serial connection that are best described as "chatty." For example, the synchronization logic may exchange numerous signals and acknowledgments over the connection in a linear fashion. In particular, synchronization logic at one end of the connection may specifically request (e.g., read) records, one at a time, from the client dataset at the other end of the connection. The synchronization logic waits until one requested record is received before requesting the next record.

Detailed Description Text (22):

It is worth noting that the above-described data loss would be caused in part by the unquestioning making of inbound changes at the client dataset. And yet, the unquestioning making of such changes is otherwise an attractive feature that helps to largely confine synchronization logic (e.g., conflict-resolution logic) to the synchronizer's core. Largely confining the synchronization logic to the synchronizer's core permits the client-specific accessor logic for each client dataset to be relatively simple, so that even simple client datasets or devices (e.g., devices without real-time clocks) can be capable of supporting adequate accessor logic. This ability to work even with simple datasets and devices is worth preserving. Therefore, what is also needed are synchronization technique(s) that continue to be suitable for synchronizing even simple client datasets or devices, even as the technique(s) also allow a client dataset to be changed during a synchronization without permitting errors such as loss of wanted data.

Detailed Description Text (27):

D. Low Reliability or Inconsistent Availability

Detailed Description Text (28):

Certain communication systems have low reliability or inconsistent availability. Low reliability may take the form of, for example, lost messages or spontaneous disconnection. Early versions of traditional, session-based synchronization (e.g., over a serial connection), dealt with communication errors or failure, if at all, simply by breaking the connection (if it has not already been broken) and aborting the synchronization. Thereafter, the user may initiate a new, full synchronization session, hopefully to its completion. (Starfish has developed a robust technique that improves upon the traditional, session-based synchronization scheme by being able, in the new synchronization session, to avoid repeating outbound transmissions that were already performed in the first session before it was aborted. This technique is further

described for example in the incorporated U.S. patent application having Ser. No. 09/208,815.)

Detailed Description Text (31):

In some communication systems, especially message-based or connectionless systems, there may not be a guarantee that transmissions will be delivered in FIFO order. This inability to guarantee order can cause a variety of problems. Such problems are exacerbated if there is a design goal to keep the client dataset's accessor logic relatively simple as compared to the synchronizer core's logic to better support simple client datasets or devices. Therefore, what is needed are synchronization technique(s) that can properly function even across communication systems that do not guarantee FIFO delivery and even for relatively simple client datasets or devices (such as client datasets and devices without real-time clocks).

Detailed Description Text (33):

In certain communication systems, characteristics such as bandwidth, cost, latency, availability, and the like are different between the outbound and inbound directions with respect to a client dataset. Similarly, communication characteristics may differ based on the time of day or other factors. Such differences are another reason why users need the more flexible synchronization transactions envisioned in the previous discussions. Therefore, what is needed are synchronization technique(s) that can be selective about which transactions to execute under which circumstances to obtain better overall performance and cost-effectiveness.

Detailed Description Text (87):

The nominally two-ply synchronization method 580 differs from the four-ply synchronization method 570 in that the last two plies 573 and 574 of one synchronization are combined with the first two plies 571 and 572, respectively, of a next synchronization, to form the nominally two-ply synchronization method 580. This is a form of pipelining the four-ply synchronization method 570 to provide higher synchronization throughput. Put another way, this is a way of starting a new synchronization before the previous synchronization has fully ended, i.e., before all exchanged records of the previous synchronization have been fully mapped to each other in the server's internal database. The nominally two-ply synchronization method 580 can be advantageously employed to repeatedly synchronize (e.g., perpetually synchronize) a dataset that undergoes frequent changes. For example, a client may be programmed to check for fresh client changes or acknowledgment requests (e.g., requests for mappings) according to a user-settable schedule (e.g., every N minutes, where N is user-settable and defaults, e.g., to thirty), and if such fresh changes or requested acknowledgments are found, the client initiates the method 580 by sending the transmissions of the first ply 581. For another example, a client may be programmed to initiate the method 580 as soon as it detects that the server has become reachable via the communication medium, e.g., as soon as the server has come into transmission range.

Detailed Description Text (90):

Further, these free-form transactions are performed without needing to lock the client dataset against user modification for any user-appreciable lengths of time (; e.g., any processing delays are much shorter than the maximum possible transmission latency encountered by synchronization signals.) Still further, these free-form transactions are performed even if the client is a simple one that does not ordinarily perform conflict resolution locally but relies on the server to provide conflict resolution. These free form transactions may be controlled by user input to the synchronization system (e.g., to the server or to a client accessor) for initiating synchronization. The user input may specify transactions (e.g., "send changes", "get changes", "synchronize") either for immediate execution or for timed execution (e.g., "synchronize at 2:00 am"). Such free-form transactions may be implemented for example by freely undertaking the "plies" of transmissions in the methods 500, 570, or 580 in an asynchronous or connectionless manner.

Detailed Description Text (92):

During the synchronization process, dataset changes and other signals are sent between a client (dataset accessor) and the server (synchronizer core). As is described elsewhere in this document, these changes/signals may be sent in multiple batches. Due to the limitations of the communication system used, some batches may get lost. A solution to this problem according to the present invention involves seeking acknowledgment from the other side for sent (batches of) changes/signals. However, in a high-latency environment, it is not desirable to await acknowledgment of each batch before sending the next batch. Therefore, as will be further explained, preferred synchronization methods according to the present invention ensure robustness against

lost changes/signals by using acknowledgments, but do not generally require a linear, in-sequence confirmation of each batch of communications before other synchronization communications can be sent or processed. Rather, a (less-chatty) asynchronous, pipelined communication/synchronization scheme is used, in which robustness against lost changes/signals is still maintained. The methods 500, 570, and 580 embody this design approach.

Detailed Description Text (99):

Datasets are collections of data. According to the present invention, the purpose of synchronizing two, or more than two, datasets is to update them as necessary with data from one another so that they contain the same or equivalent data (generally, the latest data), at least in the portions of the datasets that the user has designated for synchronization. Each dataset may be organized into individual data records. For example, a dataset having contact information may be organized into records, including a record listing a "Bill Smith's" phone numbers and addresses and another record listing a "Ted Brown's" phone numbers and addresses. In general, if records have been added to any dataset before a synchronization, then equivalent records are added to the other datasets as a result of the synchronization. Also, generally, if modifications or deletions of records have been made to one dataset before the synchronization, then equivalent modifications and deletions of corresponding records are made to the other datasets as a result of the synchronization. (The preceding discussion of synchronization according to the present invention becomes more complicated if conflicts or duplicates are present. Conflicts and duplicates are further described in later sections.)

Detailed Description Text (101):

In synchronizing two, or more than two, datasets, a correspondence is generally established between particular records across the datasets. For example, a contact record for "Bob Smith, of Acme Widgets" may exist in every dataset (perhaps as a result of synchronization), and these records in different datasets may correspond to one another. The records in a dataset may be of various data types, for example, a time-zone type, a contact type, a calendar-entry type, a task (or "to do"-list-entry) type, a memo type, an electronic-mail type, or other types. In general, each record may include data organized into one or more data fields. For example, a contact-type record may include data for a "last name" field, a "first name" field, a "company" field, and many other fields. For many typical data types, it is not necessary for each record of the data type to have data for every possible field. For synchronization, a correspondence is typically established between particular data fields across datasets. For example, a "title" field for contact records in one dataset may correspond to a "Job Title" field for contact records in another dataset. In general, the systems and methodologies of the present invention can be adapted to work with any one type of data, or with any multiple types of data, and with arbitrarily defined or named data fields.

Detailed Description Text (104):

When performing synchronization, a synchronization system transforms records from one dataset's representation into another dataset's representation. For example, the system may transform from an Internet Sidekick.RTM. cardfile for business contacts into a synchronization-system-internal representation. Typically, there is a one-to-one relationship between records in the source and target datasets. If this is not the case, however, the component of the system that interacts with a non-conforming dataset (e.g., a dataset accessor) includes logic to handle this non-conformity.

Detailed Description Text (111):

Occasionally, the user may cause the same, or matching, information to exist in different datasets without using the present invention, and then use the present invention to synchronize the datasets. For example, the user may cause records to exist for a "Bob Smith, of Acme Widgets" in multiple datasets, either by adding such records or by modifying existing records into such records. If the definition of the contact data type requires that the first name, last name, and company information for each contact be unique, then the example records would by definition match one another. In such a situation, simpleminded propagation of each added or modified record in each dataset to all other datasets would result in a duplication of records. Therefore, the present invention performs duplicate resolution to prevent such duplication. Automatic and user-assisted methods for resolving duplicates according to the present invention are discussed in later sections. Preferably, logic for performing duplicate or conflict resolution is confined as much as possible to the synchronization system's core and not at the client datasets' accessor.

Detailed Description Text (115):

Some information devices may be so simple as not to have real-time clocks (having time and date) at all. Datasets in some of such devices and other devices, instead, use simple counters as their (non-real-time) clocks. Such counters are incremented either continually at unspecified and possibly non-constant rates or upon the occurrence of particular activities or events. For example, the counter may be incremented whenever any record is changed in the dataset (such that the counter counts changes). Counter values may be suitable for use as timestamps (e.g., last-modification timestamps) for the dataset, especially if the counters are guaranteed to increase between one change to any record within the dataset and any subsequent change to any record within the dataset. However, such timestamps for the dataset cannot readily be converted into a real time (e.g., GMT time and date) and therefore cannot readily be compared with timestamps (e.g., real-time timestamps) from other datasets (and therefore are not globally-usable). Synchronization schemes according to embodiments of the present invention will work with even such datasets. In particular, the synchronizer, and especially the client accessor, does not need to compare timestamps from such a client dataset's clock with timestamps from another clock, such as the reference dataset's clock. For data from such datasets, however, timestamp-based automatic conflict resolution (e.g., "latest change wins") may not be possible, as will be described. Regardless, even for client datasets with real-time clocks that produce globally-usable timestamps, comparison between timestamps from different devices can be problematic due to time-drift and other issues, and when needed are preferably confined to the synchronization system's core. Preferably, all client datasets do have and use real-time clocks that provide globally-usable timestamps (e.g., convertible into GMT). Most especially, the synchronizer core (including the reference dataset) preferably has and uses a real-time clock that provides globally-usable (e.g., convertible into GMT) timestamps for the core's records and operations.

Detailed Description Text (118):

FIG. 6 is a block diagram that shows the architecture of a synchronizer 600 according to an embodiment of the invention. The synchronizer 600 includes a client accessor 605 and a synchronizer core 240A which communicate with each other by exchanging messages called "action objects" 615, 617, 619, 621, 623, and 625 via a communication network 610 that is found in the environment of the synchronizer and via a communication layer 630 and input queue 632 and output queue 634. In the preferred embodiment, the synchronizer 600 includes a synchronization server, capable of wireless-and Internet-based communication, on which the synchronizer core 240A is implemented.

Detailed Description Text (119):

The communication network 610 may include wireless or wireline networks for providing point-to-point, real-time connections (e.g., sockets, full- or half-duplex connections, and the like, etc.) and/or wireless or wireline networks for transferring messages (e.g., paging messages or e-mails). Examples of suitable communication networks include GSM networks (Global System for Mobile Communications), PCS networks (Personal Communication Services), TDMA (Time-Division Multiple Access) networks, CDMA (Code-Division Multiple Access) networks, wireless paging networks, AMPS (Advanced Mobile Phone System) networks, CDPD networks (Cellular Digital Packet Data), POTS networks (Plain-Old-Telephone-Service), the Internet, other IP-based (Internet-Protocol) or TCP/IP-based (Transmission Control Protocol/IP) networks, networks supporting SMS (Short Messaging Service), networks supporting WAP (Wireless Application Protocol) or HDTP (Handheld Device Transport Protocol), other cellular networks, other paging networks, other digital or analog wireless networks, other wireline networks, other circuit- or packet-based networks, and like networks. (WAP specifications are available for free download on the Internet at <http://www.wapforum.com/docs/technical.htm>.)

Detailed Description Text (120):

The client accessor 605 handles access to information in a particular client dataset (not shown) and handles delivery and receipt of information across the communication channel 610 on the client side. The communication layer 630 handles delivery and receipt of information across the channel 610 on the synchronizer core's side. The communication layer 630 includes a listener 635 for receiving and unpackaging (e.g., deserializing) information from the communication channel 610 into an input queue 632. The communication layer 630 also includes a writer 640 for packaging (e.g., serializing) and sending information onto the communication channel 610 from the output queue 634.

Detailed Description Text (121):

The synchronizer core 240A includes a GUD 255A that stores a super-set of information

from all other datasets as a central repository of information, including not only the latest, conflict-resolved values of data records but also status information, including the correspondence, or mapping, of records in the GUD to records in other datasets. The synchronizer core also includes a synchronizer engine 254A that controls and maintains the GUD and works with all client accessor(s) to collectively drive and control the synchronization process. The Synchronization engine works with client accessor(s) by receiving action objects via the input queue and sending action objects via the output queue.

Detailed Description Text (122):

Preferably, the client accessor 605 exists as a software process that runs on a same device as the particular client dataset. Preferably, the listener 635 and the writer 640 exist as individual software processes that respectively replenish and withdraw action objects from the in queue and out queue, respectively, when available. The action objects 619, 621, 615, and 625 that are used by the synchronizer core 240A and the accessor 605 for mutual communicating will be further described in a later section. For now, it is sufficient to mention that action objects are a defined format for conveying dataset changes and other information between the synchronizer core and client accessors, in either direction. The action objects 615, 617, 623, and 625 are action objects that have been packaged (for example, serialized) for transport across the communication channel 610.

Detailed Description Text (133):

FIG. 7C is a table that shows portions of the mapping table 707 (from FIG. 7A) that includes status information relating to a particular client. The mapping table 707 includes status information 719 about the client as a whole, for example its status, capabilities, limitations, and preferences as will be further described in later sections. The mapping table 707 also contains status information about each individual record last known by the synchronizer to be in the client. In FIG. 7C, each row 720 of the mapping table 707 corresponds to one particular GUD record. If a GUD record has a corresponding record in the client, then the GUD record's row in the mapping table will include both the GUD record's GUD ID and its client record's client ID 722, to thereby map the two ID's to each another. (A client ID uniquely identifies a client record within the client or client accessor.) If no record in the client corresponds to the GUD record, then the GUD record's row in the mapping table will be nonexistent or empty.

Detailed Description Text (144):

C. Client Dataset and Accessor

Detailed Description Text (145):

FIG. 8 is a table that shows client records 802, including (user) data 803 and some status information 804, that are preferably available to the client accessor. In general, different client datasets are free to and are expected to store their records in possibly different formats and using possibly different storage technologies from one another. The accessor for each client includes the client-specific knowledge for accessing client records and transferring their contents or status as necessary to the synchronizer core using action objects. In FIG. 8, each row 808 of the table represents a client record. The status information 804 preferably includes information for determining which client records are fresh with respect to the GUD; i.e., which records include values that the GUD may not have seen before. The status information 804 also preferably includes information for determining whether changes received from the GUD/server are obsolete (e.g., superseded). More particularly, according to the preferred embodiment, the status information 804 includes for each client record a unique client record identifier 812 ("client ID"), client last-modified timestamp information 814, a last-sent-and-ack'd (acknowledged) timestamp (e.g., counter) 827, a last-GUD-sent timestamp 828, a GUD-send-last-processed timestamp 829, and other information 830.

Detailed Description Text (146):

Each client ID 812 uniquely identifies a client record within the client accessor. In the preferred embodiment, each client ID is an integer having at least 32 bits. The client record ID may be referred to as cr_ID.

Detailed Description Text (151):

Note that most client datasets typically maintain the field cmT_Mod natively in the dataset itself. For legacy client datasets not designed with the present invention in mind, the client accessor is responsible for separately maintaining the fields csT_SentAckd, gsT_GudSent, and cmT_ProcGudSent, perhaps in the client accessor's own

assigned storage area (e.g., persistent storage), for use in synchronization. This is a means of retrofitting a legacy client device to work with the present invention. Note that according to the present invention, during ordinary, non-synchronization use of the client dataset (e.g., by the user), the fields csT_SentAckd, gsT_GudSent, and cmT_ProcGudSent are not needed. Instead, these fields need only be used by the client accessor for synchronization.

Detailed Description Text (154):

Synchronization methods of the present invention communicate via synchronization messages called action objects. Action objects are used to communicate dataset changes and other signals. The meanings and usages of action objects according to the present invention are not limited to a chatty, single-session synchronization context. Action objects relevant to synchronization according to the present invention include the following types, which will be further described: Action Insert Record (C-Dominant Version) Action Update Record (C-Dominant, C-Untested, and (optionally) C-Conflicting Versions) Action Delete Record (C-Dominant, C-Untested, and (optionally) C-Conflicting Versions) Action Existing Record List (C-Untested Version) Action Request Ack Records Action Ack Records Action Retrieve Records Action Update Map Action Request Ack Maps Action Ack Maps Action Backup Record (Optional) Action Last Backup Sent (Optional) Action Retrieve Backup (Optional)

Detailed Description Text (157):

In general, a timestamp is either a record modification time or a synchronization communication time (e.g., send time). Preferably, each dataset (client or GUD) uses a same clock (e.g., a dataset-specific counter) to generate modification timestamps and timestamps of synchronization communications or activities. In this way, modification timestamps and synchronization times from the same dataset (or its accessor) can be compared.

Detailed Description Text (185):

The client determines modified records and added records, for example, as follows: 1) if the client keeps last-modification timestamps (e.g., counter values) of records (preferable), then if the last-modification time of any record exceeds the time of the record's last synchronization with the server, then that record is either an added or updated record (the client doesn't need to worry exactly which type it is). 2) Otherwise (the client is of the non-preferred type that does not keep last-modification timestamps), the client will send CRC-type values (Cyclic Redundancy Check-type values) corresponding to the entire record for all records to the server so that the server can compare with the CRC-type values that were recorded for the client's records at the end of the previous synchronization of the record to quickly determine which have changed. Such CRC-type values may be sent via Action Update Record (C-Untested Version) action objects (e.g., along with all data for all records). Alternatively, such CRC-type values may be sent via an Action Existing Record List (C-Untested Version) action object (e.g., without all data for the records), whereupon the server can determine the changed records and specifically request them from the client using an Action Retrieve Records action object that specifically identifies and requests the changed client records.

Detailed Description Text (189):

The client can request fresh (with respect to the client) changes from the server at any time. The sender (e.g., client) of such a request sends an Action Retrieve Records action object to the recipient (e.g., server). The Action Retrieve Records action object may request all fresh changes (default) or specify a list of client records for which fresh changes, if any exist, are desired. The client's requesting 532A of changes from the server in the method 500 of FIG. 5A is exemplary.

Detailed Description Text (220):

As previously mentioned, it is preferred that all datasets ever synchronized by the synchronizer provide globally-usable timestamps for their records and for their communications, such that the server can always perform automatic, timestamp-based (e.g., latest change wins) conflict resolution. Under this scenario, c-conflicting changes need not ever be sent, and preferably are not ever used. However, if the server detects a conflict but cannot compare timestamps in order to perform timestamp-based conflict resolution, it will prompt the user to make a choice between conflicting changes. Optionally, the server can detect a conflict and use the action object framework itself to delegate the user-assisted conflict-resolution task to a client accessor. This delegation is particularly useful if the server cannot otherwise present a user interface to the user--i.e., if the user is at the client and synchronizing wirelessly and has no browser open to the server.